

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-309231
(43)Date of publication of application : 04.11.1994

(51)Int.Cl. G06F 12/08
G06F 12/08

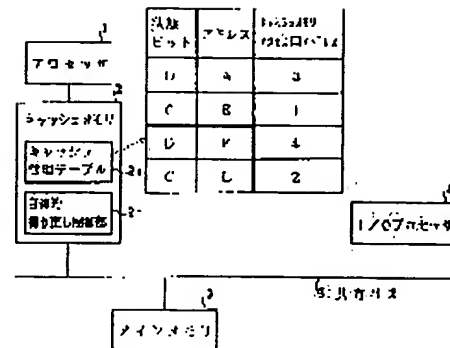
(21)Application number : 05-122191 (71)Applicant : AGENCY OF IND SCIENCE & TECHNOL
(22)Date of filing : 27.04.1993 (72)Inventor : SATO MASATOSHI

(54) CACHE MEMORY CONTROL METHOD

(57)Abstract:

PURPOSE: To reduce the traffic from a cache memory to a main memory and to improve the performance of the whole system.

CONSTITUTION: The write-back type cache memory 2 is provided with a writing-back process only for writing data back from the cache memory 2 to the main memory 3 separately from a writing-back process for expelling data from the cache memory 2 to the main memory. In this process, it is decided whether or not data of one entry of the cache memory 2 is dirty and when so the data are written back to the main memory 3. It is confirmed that the writing-back process normally ends and the data in the cache memory 2 do not change, the cache state is cleaned. If there is a request to occupy a bus from another access means during this process, on the other hand, the process is refused.



LEGAL STATUS

[Date of request for examination] 27.04.1993
[Date of sending the examiner's decision of rejection]
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number] 2099151
[Date of registration] 22.10.1996
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

Copyright (C): 1998,2003 Japan Patent Office

Japanese Laid-Open Patent Application No. 6-309231 Publication

[0018] Next, the autonomous write back operation shown in FIG. 1 shall be explained. First, this autonomous write back operation is assumed to be activated at a predetermined cycle. Moreover, this cycle is experimentally obtained in advance according to the system. Now, the autonomous write back control unit 2b activates, and the write back operation is started. With this, the autonomous write back control unit 2b first searches the cache entry management table 2a and judges whether the condition bit of a particular entry is dirty (D) or clean (C) (step S1). Moreover, here, dirty refers to the case where the data inside the cache memory 2 and the data inside the main memory 3 do not match, in other words, the case where a write operation has been performed on the data of the cache memory 2 and the write back operation to the main memory 3 is not carried out, and a write back is necessary. Furthermore, clean refers to the case where the data inside the cache memory 2 and the data inside the main memory 3 match, in other words, the case where there is no need for the data in the cache memory 2 to be written back to the main memory 3.

[0019] In step 1, in the case where the condition bit is dirty, it is checked that the shared bus 5 is free (step S2) and, as the next autonomous write operation, a bus command is generated and sent (step S3).

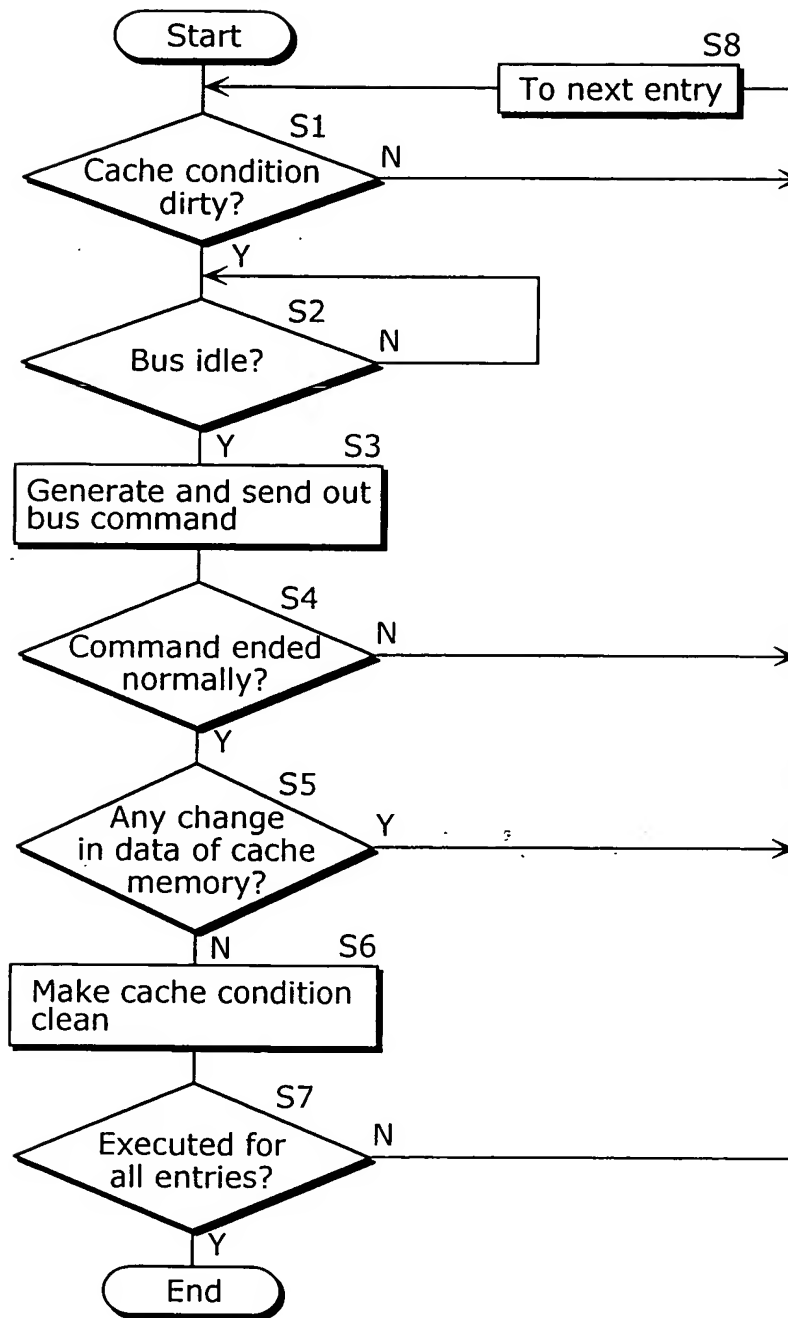
[0020] FIG. 4 shows the format of the bus command. More specifically, the bus command is made up of a control flag indicating whether or not it is a command for the autonomous write back, an address of the main memory where the write back is to be performed, write back data, and so on. Furthermore, the control flag is set to indicate "0" in the case of the autonomous write back and "1" in the case of the normal write back; and becomes a command allowing midstream abortion in the case of the autonomous write back operation.

[0021] In step S3, upon starting the autonomous write back, the autonomous write back control unit 2b waits for the end of such process (step S4). In other words, it is judged whether or not a command end notification for the sending of the bus command is received. In step S4, in the case where the command ends normally, it is judged whether or not the write back data of the cache memory 2 has changed (step S5). In the case where it has not changed, it is assumed that the condition bit of the cache management

table 2a is a bit indicating clean, as the data of the cache memory and the data of the main memory 3 match (step S6). More specifically, during the execution of the autonomous write up operation, it is judged whether or not the processor 1 has rewritten the data, and the cache memory 2 is assumed as being in a clean condition by verifying that the data has not been rewritten. [0022] In addition, in step S6, when the corresponding entry in the cache management table 2a is in a clean condition, it is judged whether or not the aforementioned process has been executed on all the entries (step S7). In the case where the process has been executed on all of the entries, the autonomous write back operation ends. In the case where there is a remaining entry, the processes advances to the next entry (step S8), returns to step S1, and the aforementioned operation is repeated.

[0023] Furthermore, when the condition of the cache is clean in the aforementioned step S1; when the command does not end normally in step S4; and when the data of the cache entry is rewritten in step S5, the process moves to step S8, and the next processing of the next entry is performed.

FIG. 1



Flowchart of the first embodiment of the method in the present invention

FIG. 4

Control flag	Address	Data	-----
-----------------	---------	------	-------

Write back bus command in the method in the present invention

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平6-309231

(43)公開日 平成6年(1994)11月4日

(51)IntCl.⁵

G 0 6 F 12/08

識別記号

3 1 0 Z

庁内整理番号

7608-5B

F I

技術表示箇所

J 7608-5B

審査請求 有 請求項の数 1 F D (全 10 頁)

(21)出願番号 特願平5-122191

(22)出願日 平成5年(1993)4月27日

(71)出願人 000001144

工業技術院長

東京都千代田区霞が関1丁目3番1号

(72)発明者 佐藤 正俊

東京都港区虎ノ門1丁目7番12号 沖電気

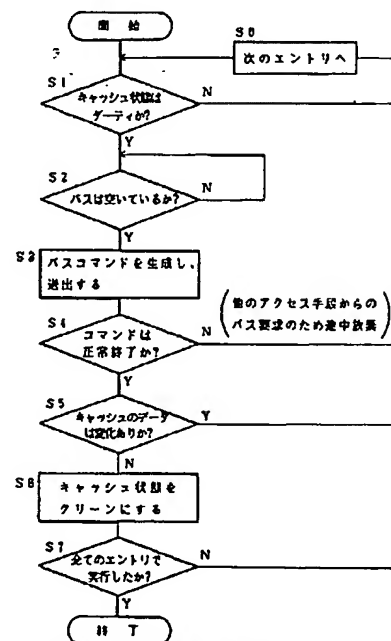
工業株式会 社内

(54)【発明の名称】 キャッシュメモリ制御方法

(57)【要約】

【構成】 ライトバック方式のキャッシュメモリにおいて、キャッシュメモリからデータを追い出すためのメインメモリへの書き戻し処理とは別に、キャッシュメモリのデータをメインメモリに書き戻すだけのための書き戻し処理を設ける。この処理は、先ずキャッシュメモリの一つのエントリのデータがダーティであるか否かを判定し、ダーティであった場合はそのデータをメインメモリに書き戻す。そして、書き戻しが正常終了し、かつキャッシュメモリ上のデータが変化していないことを確認し、キャッシュ状態をクリーンとする。一方、この処理中に他のアクセス手段からバスの占有要求があった場合、当該処理は放棄される。

【効果】 キャッシュメモリからメインメモリへのトラフィックが減少し、システム全体の性能を向上させることができる。



本発明方法における第1の実施例のフローチャート

【特許請求の範囲】

【請求項1】 キャッシュメモリにメインメモリからデータが読込まれる際に、当該キャッシュメモリから追い出されるデータと、このデータに対応する前記メインメモリのデータとが一致しない場合、当該データを、前記キャッシュメモリと前記メインメモリが接続される共有バスを介して前記メインメモリに書き戻すキャッシュメモリ制御方法において、

前記共有バスが空いている場合、前記キャッシュメモリからのデータ追い出し時における書き戻し処理とは別に、前記メインメモリのデータと一致していない前記キャッシュメモリのデータを書き戻すだけの自律的書き戻し処理を行い、

前記自律的書き戻し処理の実行中に、他のアクセス手段から前記共有バスの占有要求が生じた場合は、前記自律的書き戻し処理を中断することを特徴とするキャッシュメモリ制御方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、ライトバック方式のキャッシュメモリにおいて、システムとしての性能向上を図ることのできるキャッシュメモリ制御方法に関する。

【0002】

【従来の技術】 計算機システムにおけるキャッシュメモリは、システムにおける処理の高速化を図るために、重要な技術であり、その実現方式には多くの方式がある。そして、そのキャッシュメモリからメインメモリへのデータの書き戻し方法に関しては、代表的な二つの方法がある。

【0003】 この二つの方法とは、ライトスルー方式とライトバック方式である。まず、ライトスルー方式とは、プロセッサからのデータの書込み時に、メインメモリとキャッシュメモリとの両方にそのデータを書き込む方式である。この方式は、メインメモリとキャッシュメモリの一貫性が保て、入出力プロセッサ等、他のアクセスがメインメモリに生じる時に利点がある。一方、データ書込み時に毎回メインメモリをアクセスするため、キャッシュメモリ（プロセッサ）とメインメモリ間のトラフィックが高くなってしまい、システムの性能低下の原因となってしまう欠点がある。

【0004】 これに対し、ライトバック方式では、プロセッサの書込み時には、キャッシュメモリのデータのみが更新され、メインメモリへの書込みは、キャッシュメモリ上のデータがキャッシュメモリから追い出される時まで遅らせられ、その時点で実際の書込みが行われる。この場合、何度も書込みが生じるようなアクセスでは、キャッシュメモリがヒットしている限り、キャッシュメモリ（プロセッサ）とメインメモリ間のトラフィックは生じることがなく、ライトスルー方式のようなシステムの性能低下の原因は除去することができる。

【0005】 また、共有バスを介して複数のプロセッサがメインメモリを共有する共有メモリ型マルチプロセッサにおいても、システムにおける処理の高速化を図るため、キャッシュメモリは重要な技術となっている。この場合、キャッシュメモリはプロセッサ毎に設けられており、各キャッシュメモリからメインメモリへのデータ書き戻し方法に関しては、代表的なものとして上述したライトバック方式がある。また、このようなライトバック方式でのキャッシュメモリ間のコヒーレンスを保つ方法の代表的な例としてインバリデイトがある。以下、このようなライトバック／インバリデイトの組合せ例としてイリノイ方式を説明する。

【0006】 図2は、キャッシュメモリにおける4状態の説明図である。まず、I {Invalid} とは、そのブロックが無効であり、リプレイスの対象であることを示している。また、CE {Private Clean (Clean Exclusive)} とは、そのブロックのデータはメインメモリのデータと一致しており、かつ、唯一のコピーであることを示している。更に、CS {Shared Clean} とは、そのブロックのデータはメインメモリのデータと一致しており、かつ、同一データのコピーが他のキャッシュメモリにも存在する可能性があることを示し、DE {Dirty (Dirty Exclusive)} とは、そのブロックのデータはメインメモリのデータと一致しておらず、かつ唯一のコピーであることを示している。

【0007】 このイリノイ方式では、プロセッサの書込み時には、キャッシュメモリのみが更新され、メインメモリへの書込みは、キャッシュメモリ上のデータがキャッシュメモリから追い出される時まで遅らせられ、その時点で実際の書込みが行われる。この場合、何度も書込みが生じるようなアクセスでは、キャッシュメモリがヒットしている限り、キャッシュメモリ（プロセッサ）とメインメモリ間のトラフィックが生じることはなく、マルチプロセッサで共有される相互接続ネットワークのトラフィックを軽減している。

【0008】

【発明が解決しようとする課題】 しかしながら、上記ライトバック方式では、メインメモリへの書込みのタイミングを遅らせることにより、キャッシュメモリ（プロセッサ）とメインメモリ間のトラフィックをある程度軽減することができるが、依然としてキャッシュメモリ上のデータがキャッシュメモリから追い出される時には、キャッシュメモリ（プロセッサ）とメインメモリ間のトラフィックが生じる。そして、このようなトラフィックが、ライトスルー方式のように、バスを共有する他の処理装置からや、共有メモリ型マルチプロセッサにおける他のキャッシュメモリ（プロセッサ）からのトラフィックを妨げ、システムの性能低下を招くといった問題点が残されていた。

【0009】 本発明は、上記従来の問題点を解決するた

めになされたもので、ライトバック方式のキャッシュメモリにおいて、システムの性能低下を防止することができるキャッシュメモリ制御方法を提供することを目的とする。

【0010】

【課題を解決するための手段】本発明のキャッシュメモリ制御方法は、キャッシュメモリにメインメモリからデータが読込まれる際に、当該キャッシュメモリから追い出されるデータと、このデータに対応する前記メインメモリのデータとが一致しない場合、当該データを、前記 10 キャッシュメモリと前記メインメモリが接続される共有バスを介して前記メインメモリに書き戻すキャッシュメモリ制御方法において、前記共有バスが空いている場合、前記キャッシュメモリからのデータ追い出し時における書き戻し処理とは別に、前記メインメモリのデータと一致していない前記キャッシュメモリのデータを書き戻すだけの自律的書き戻し処理を行い、前記自律的書き戻し処理の実行中に、他のアクセス手段から前記共有バスの占有要求が生じた場合は、前記自律的書き戻し処理を中断することを特徴とするものである。

【0011】

【作用】本発明のキャッシュメモリ制御方法においては、バスラインが空いている場合、キャッシュメモリからデータを追い出すためのメインメモリへの書き戻し処理とは別に、キャッシュメモリのデータをメインメモリに書き戻すだけのための自律的書き戻し処理を行う。この自律的書き戻し処理は、まず、キャッシュメモリの一つのエントリのデータがダーティであるか否かを判定し、ダーティであった場合はそのデータをメインメモリに書き戻す。そして、書き戻しが正常終了し、かつ 30 キャッシュメモリ上のデータがプロセッサによって書き換えられていないことを確認し、キャッシュ状態をクリーンとする。一方、この自律的書き戻し処理中に他のアクセス手段からバスの占有要求があった場合、当該処理は放棄される。

【0012】

【実施例】以下、本発明の実施例を図面を用いて詳細に説明する。

《第1の実施例》図1は本発明のキャッシュメモリ制御方法における第1の実施例を示すフローチャートであるが、この説明に先立ち、その制御方法を実施するための 40 システム構成について説明する。図3に、このキャッシュメモリシステムを示す。図のシステムは、情報処理装置におけるメモリシステムを示しており、プロセッサ1、キャッシュメモリ2、メインメモリ3、I/Oプロセッサ4、共有バス5からなる。

【0013】プロセッサ1は、各種の制御を司る制御部であり、特に、メモリシステムにおいて、キャッシュメモリ2におけるデータのライト/リード制御を行うものである。キャッシュメモリ2は、その内部にキャッシュ 50

管理テーブル2aと、自律的書き戻し制御部2bとを備えている。キャッシュ管理テーブル2aは、複数のエントリ毎に、そのエントリのデータが有効か無効かを示す状態ビットと、メインメモリ3におけるアドレスと、キャッシュメモリ2におけるデータの記憶場所であるキャッシュメモリアクセス用アドレスのフィールドで構成されている。例えば、図示例では、メインメモリ3のアドレス“A”のデータがキャッシュメモリ2上のアドレス“3”に格納されており、かつ、このデータはD（ダーティ）であることを示している。

【0014】また、自律的書き戻し制御部2bは、本実施例の特徴点をなすもので、キャッシュメモリ2からメインメモリ3へのデータの書き戻しを、自律的、即ち、プロセッサ1からの指示とは無関係に行う機能を有している。

【0015】また、メインメモリ3は、プロセッサ1が実行するためのプログラムやデータを格納するためのメモリであり、共有バス5を介して格納されているデータ等がリード/ライトされる。I/Oプロセッサ4は、図 20 示省略した外部記憶装置等のI/Oの制御を行うプロセッサである。更に、共有バス5はキャッシュメモリ2やI/Oプロセッサ4等とメインメモリ3とを接続するための共有バスである。

【0016】次に、このように構成されたメモリシステムにおけるキャッシュメモリ制御方法を図1のフローチャートを参照して説明する。先ず、図1に示す自律的書き戻し処理の前に、通常のライトバック処理について説明する。

【0017】今、キャッシュメモリ2の各エントリには、全て何等かのデータが入っていると看做する。このような状態で、プロセッサ1が、例えば、あるデータに対してリードを行う場合、プロセッサ1はキャッシュ管理テーブル2aを参照し、そのリード要求のデータのアドレスがあるかどうかを調べる。そして、そのアドレスが存在した場合は、キャッシュメモリアクセス用アドレスに基づき、そのデータをキャッシュメモリ2からリードする。一方、そのアドレスが存在しなかった場合、プロセッサ1は、先ず、キャッシュメモリ2のいずれかのエントリのデータをメインメモリ3に書き戻し、その後、そのエントリにメインメモリ3からリードしたデータを格納する。

【0018】次に、図1に示す自律的書き戻し処理を説明する。先ず、この自律的書き戻し処理は、ある所定期間で起動されるとする。尚、この周期はそのシステムによって予め実験的に求めておく。今、自律的書き戻し制御部2bが起動し、書き戻し処理が開始されたとする。これにより、自律的書き戻し制御部2bは、先ず、キャッシュ管理テーブル2aを検索し、あるエントリの状態ビットがダーティ（D）かクリーン（C）かを判定する（ステップS1）。尚、ここで、ダーティとは、キャッ

5

ッシュメモリ2内のデータとメインメモリ3のデータとが不一致の場合、即ち、キャッシュメモリ2のデータに書き込み操作が行われており、かつメインメモリ3への書き戻しがされておらず、書き戻しが必要な場合である。また、クリーンとは、キャッシュメモリ2内のデータとメインメモリ3のデータとが一致している場合、即ち、キャッシュメモリ2上のデータをメインメモリ3に書き戻す必要がない場合である。

【0019】ステップS1において、状態ビットがダーティであった場合は、共有バス5が空いていることを確認し（ステップS2）、次の自律的書き戻し処理として、バスコマンドを生成し、これを送出する（ステップS3）。

【0020】図4に、バスコマンドのフォーマットを示す。即ち、バスコマンドは、自律的書き戻しのためのコマンドか否かを示す制御フラグと、書き戻しを行うメインメモリ3のアドレスと、書き戻しデータ等で構成されている。また、制御フラグは、自律的書き戻しの場合は“0”、通常の書き戻しの場合は“1”といったように設定され、自律的書き戻しの場合は、途中放棄を許すコマンドとなっている。

【0021】ステップS3において、自律的書き戻し制御部2bは、自律的書き戻し処理を開始すると、その処理の終了待ち状態となる（ステップS4）。即ち、バスコマンドの送出に対するコマンド終了通知を受けたか否かを判定する。ステップS4において、コマンドが正常終了した場合は、キャッシュメモリ2の書き戻したデータが変化していないか否かを判定し（ステップS5）、変化していない場合は、キャッシュメモリのデータとメインメモリ3のデータとが一致したとして、キャッシュ管理テーブル2aの状態ビットをクリーンを示すビットとする（ステップS6）。即ち、自律的書き戻し処理の実行中に、プロセッサ1がそのデータを書き換えたか否かを判定し、書き換えていないことを確認してキャッシュメモリ2をクリーンな状態とするものである。

【0022】そして、ステップS6において、キャッシュ管理テーブル2aの対応するエントリをクリーンな状態とすると、上記の各処理を全てのエントリで実行したか否かを判定し（ステップS7）、全てのエントリで行った場合は、自律的書き戻し処理を終了し、まだエントリが残っている場合は、次のエントリに進み（ステップS8）、ステップS1に戻って、上記の動作を繰り返す。

【0023】また、上記ステップS1においてキャッシュの状態がクリーンな場合、ステップS4においてコマンドが正常終了していない場合、およびステップS5においてそのエントリのデータが書き換えられた場合は、ステップS8に移行し、次のエントリの処理を行う。

【0024】尚、ステップS4においてコマンドが正常終了しない場合とは、例えば次のような場合である。即

6

ち、自律的書き戻しの場合のバスコマンドは、上述したように、自律的書き戻し処理によるバス占有中に、他のアクセス手段によるバス要求があった場合のバス権放棄機能を有している。このバス権放棄機能は、自律的書き戻しのバスコマンド実行中に他のバスコマンドをブロックしないための機能であり、これは、通常の書き戻しバスコマンドに、バス要求を監視し、バス要求があればバス権を放棄し、実行中の処理をリセットする（キャッシュメモリ2とメインメモリ3の両方共）機能を追加することで実現することができる。

【0025】一方、1/Oプロセッサ4は、共有バス5の状態を常時監視しており、自律的書き戻しのバスコマンド実行中は、これを認識できるようになっている。そして、この自律的書き戻し処理の実行中に制御している1/Oとのアクセス要求が発生した場合は、バス権を要求し、共有バス5を占有する。

【0026】また、書き戻しバスコマンドの途中放棄によるメインメモリ3の一部が書き換わった場合の、キャッシュメモリ2とのデータの一貫性は、全データの書き込みの完了を意味する書き戻しバスコマンドの正常終了時のみキャッシュの状態を書き換えることで維持することができる。

【0027】更に、自律的書き戻し処理中にキャッシュメモリ2上のデータが変化しているか否かの判定は、書き戻しデータをバッファしておき、このバッファしたデータと、キャッシュメモリ2上のデータとの一致検査で行う。そして、この判定と、キャッシュ状態をクリーンにする操作はアトミック操作、即ち、他の処理の影響を受けない最小の単位での処理で行う。

【0028】このように、上記第1の実施例では、通常の書き戻し処理とは別に、自律的書き戻し処理を行うようにしたので、キャッシュメモリ2がダーティな状態であることが少なくなり、従って、従来のようなデータをキャッシュメモリ2から追い出す際の書き戻し処理も殆ど不要となり、その結果、キャッシュシステムとしてのバス占有時間を減少させることができ、システムとしての性能向上を図ることができる。

【0029】《第2の実施例》図5は、本発明のキャッシュメモリ制御方法における第2の実施例を実施するためのキャッシュメモリシステムの構成図である。図のシステムは、共有メモリ型マルチプロセッサシステムにおけるメモリシステムを示しており、複数のプロセッサエレメント11（11-1～11-n）とメインメモリ12と、共有バス13とからなる。プロセッサエレメント11は、各々プロセッサ14（14-1～14-n）と、キャッシュメモリ15（15-1～15-n）とからなる。尚、プロセッサエレメント11-2～11-nの構成は、プロセッサエレメント11-1と同様であるため、内部の構成は図示省略している。また、各プロセッサエレメント11に共通の構成は、プロセッサ14、

キャッシュメモリ15として説明する。

【0030】プロセッサ14は、各種の制御を司る制御部であり、特に、共有メモリ型マルチプロセッサシステムにおいて、キャッシュメモリ15とメインメモリ12間のデータのライト／リード制御や各キャッシュメモリ15間のデータ転送等の制御を行うものである。キャッシュメモリ15は、上記第1の実施例と同様、その内部にキャッシュ管理テーブル15a(15a-1~15a-n)と、自律的書き戻し制御部15b(15b-1~15b-n)とを備えている。

【0031】キャッシュ管理テーブル15aは、複数のエントリ毎に、そのエントリのデータの状態を示す状態ビットと、メインメモリ12におけるアドレスと、キャッシュメモリ15におけるデータの記憶場所であるキャッシュメモリアクセス用アドレスのフィールドで構成されている。例えば、図示例では、メインメモリ12のアドレス“B”のデータがキャッシュメモリ15上のアドレス“1”に格納されており、かつ、このデータはCEであることを示している。尚、図中の状態ビットI、C、E、CS、DEは、従来の技術で説明した図2のキャッシュメモリ15における4つの状態を示している。

【0032】自律的書き戻し制御部15bは、本実施例の特徴点をなすもので、キャッシュメモリ15からメインメモリ12へのデータの書き戻しを、自律的、即ち、プロセッサ14からの指示による通常のライトバック処理とは無関係に行うと共に、他のキャッシュメモリ15の自律的書き戻し制御部15bと協調してこの自律的書き戻し処理を行う機能を有している。尚、ここで、協調して処理を行うとは、特定のキャッシュメモリ15のみが書き戻し処理を行うのではなく、各キャッシュメモリ15が順次処理を実行するといったことである。

【0033】また、メインメモリ12は、各プロセッサ14が実行するためのプログラムやデータを格納するための各プロセッサエレメント11が共有するメモリであり、共有バス13を介して格納されているデータ等がリード／ライトされる。尚、共有バス13には、上記第1の実施例と同様にI/Oプロセッサ等、他のアクセス手段が接続されているが、その図示は省略している。

【0034】次に、このように構成されたメモリシステムにおけるキャッシュメモリ制御方法を説明する。尚、通常のライトバック処理については、下記に示すように、上記第1の実施例と同様に行われる。即ち、今、キャッシュメモリ15の各エントリには、全て何等かのデータが入っているとする。このような状態で、プロセッサ14が、例えば、あるデータに対してリードを行う場合、プロセッサ14はキャッシュ管理テーブル15aを参照し、そのリード要求のデータのアドレスがあるかどうかを調べる。そして、そのアドレスが存在し、かつキャッシュ状態がI(無効)でない場合は、キャッシュメモリアクセス用アドレスに基づき、そのデータをキャッ

シュメモリ15からリードする。

【0035】一方、そのアドレスが存在しなかった場合、プロセッサ14は、メインメモリ12にアクセスし、そして、そのデータを、キャッシュメモリ15のキャッシュ状態がIであるエントリに格納する。また、キャッシュ状態がIのエントリがなかった場合は、FIFO(先入れ先出し)法やLRU(least recently used)法等によってキャッシュメモリ15から追い出すブロックを決定する。この場合、これがDEであった場合、先ずそのデータをメインメモリに書き戻し、その後、メインメモリからリードを行う。尚、各キャッシュメモリ15間のデータの coherence の保持に関しては、既知の方法と同様であるため、ここでの説明は省略する。

【0036】次に、本実施例における自律的書き戻し処理を説明する。図6はそのフローチャートである。先ず、この自律的書き戻し処理は、ある所定期間で起動され、プロセッサエレメント11-1のキャッシュメモリ15-1で1エントリが終了すると、次のプロセッサエレメント11-2のキャッシュメモリ15-2といったように、各キャッシュメモリ15毎に順次行い、最後のキャッシュメモリ15-nの1エントリが終了すると、最初のキャッシュメモリ15-1の2番目のエントリに進むといったようにラウンドロビン方式で行う。尚、この自律的書き戻し処理の起動周期はそのシステムによって予め実験的に求めておく。

【0037】今、自律的書き戻し制御部15bが起動し、書き戻し処理が開始されたとする。これにより、自律的書き戻し制御部15bは、先ず、キャッシュ管理テーブル15aを検索し、最初のエントリの状態ビットがダーティ(DE)かクリーン(CEまたはCS)かを判定する(ステップS1)。ステップS1において、状態ビットがダーティであった場合は、共有バス13が空いていることを確認し(ステップS2)、次の自律的書き戻し処理として、バスコマンドを生成し、これを送出する(ステップS3)。尚、そのバスコマンドのフォーマットについては、上記第1の実施例における図4で示したのと同様に、制御フラグ、アドレスおよび書き戻しデータ等で構成されている。

【0038】ステップS3において、自律的書き戻し制御部15bは、自律的書き戻し処理を開始すると、その処理の終了待ち状態となる(ステップS4)。即ち、バスコマンドの送出に対するコマンド終了通知を受けたか否かを判定する。ステップS4において、コマンドが正常終了した場合は、キャッシュメモリ15の書き戻したデータが変化していないか否かを判定し(ステップS5)、変化していない場合は、キャッシュメモリ15のデータとメインメモリ12のデータとが一致したとして、キャッシュ管理テーブル15aの状態ビットをCEとする(ステップS6)。即ち、自律的書き戻し処理の

実行中に、プロセッサ14がそのデータを書き換えたか否かを判定し、書き換えていないことを確認してキャッシュメモリ15をクリーンな状態とするものである。

【0039】そして、ステップS6において、キャッシュ管理テーブル15aの対応するエントリをクリーンな状態とすると、全てのエントリで上記処理を実行したか否かを判定し（ステップS7）、全てのエントリで行った場合は、そのキャッシュメモリ15における自律的書き戻し処理を終了する。一方、ステップS7において、全てのエントリを実行していない場合、そのキャッシュメモリ15は、自律的書き戻し処理の割り当て待ちとなる（ステップS8）。即ち、各キャッシュメモリ15の自律的書き戻し制御部15bは、各々共有バス13を監視しており、一つのキャッシュメモリ15の自律的書き戻し処理の終了をタイミングとして、次のキャッシュメモリ15の自律的書き戻し処理に移行する。

【0040】このようにして、順次各キャッシュメモリ15の自律的書き戻し処理を行い、ステップS8において、割り当て待ちを行った後、再度自キャッシュメモリ15に割り当てられると、自律的書き戻し制御部15bは、次のエントリに進み（ステップS9）、ステップS1に戻って上記の動作を繰り返す。

【0041】また、上記ステップS1においてキャッシュの状態がクリーンな場合、ステップS4においてコマンドが正常終了していない場合、およびステップS5においてそのエントリのデータが書換えられた場合は、ステップS8に移行し、割り当て待ちとなる。

【0042】ここで、上記ステップS4においてコマンドが正常終了しない場合とは、例えば次のような場合である。即ち、自律的書き戻しの場合のバスコマンドは、上記第1の実施例と同様に、自律的書き戻し処理によるバス占有中に、他のプロセッサエレメントからのライトバック動作や他の処理装置からのアクセス要求があった場合のバス極放棄機能を有している。このバス極放棄機能は、自律的書き戻しのバスコマンド実行中に他のバスコマンドをブロックしないための機能であり、これは、通常の書き戻しバスコマンドに、バス要求を監視し、バス要求があればバス極を放棄し、実行中の処理をリセットする（キャッシュメモリ15とメインメモリ12の両方共）機能を追加することで実現することができる。

【0043】また、書き戻しバスコマンドの途中放棄によるメインメモリ12の一部が書き換わった場合、キャッシュメモリ15とのデータの一貫性は、全データの書込みの完了を意味する書き戻しバスコマンドの正常終了時のみキャッシュの状態を書き換えることで維持することができる。

【0044】更に、自律的書き戻し処理中にキャッシュメモリ15上のデータが変化しているか否かの判定は、書き戻しデータをバッファしておき、このバッファしたデータと、キャッシュメモリ15上のデータとの一致検

査で行う。そして、この判定と、キャッシュ状態をクリーンにする操作はアトミック操作、即ち、他の処理の影響を受けない最小の単位での処理で行う。

【0045】尚、上記第2の実施例では、それぞれのプロセッサエレメント11-1～11-nのキャッシュメモリ15が、その自律的書き戻し処理を協調的に行う方法として、各キャッシュメモリ15を1エントリずつ行ったが、これに限定されるものではなく、この他にも、例えば、キャッシュメモリ15-1の全エントリが終了したら、次のキャッシュメモリ15-2といったように、各キャッシュメモリ15の全エントリずつ行うよう構成してもよい。

【0046】このように、上記第2の実施例では、通常の書き戻し処理とは別に、自律的な書き戻し処理を行うようにしたので、キャッシュメモリ15のデータがダーティな状態であることが少なくなり、従って、従来のようなデータをキャッシュメモリ15から追い出す際の書き戻し処理も殆ど不要となり、キャッシュシステムとしてのバス占有時間を減少させることができる。また、その自律的書き戻し処理を他のキャッシュメモリ15と協調的に行うようにしたので、各キャッシュメモリ15間でダーティな状態のデータの偏りもなく、マルチプロセッサシステムとしての性能向上を図ることができる。

【0047】

【発明の効果】以上説明したように、本発明のキャッシュメモリ制御方法によれば、通常の書き戻しとは別に、バスの空いている時に、キャッシュメモリ上のデータを書き戻すだけの自律的書き戻し処理を行い、かつ、この処理を行っている途中で他のアクセス手段から共有バスの占有要求があった場合は、当該処理を放棄するようにしたので、キャッシュメモリ上のデータがキャッシュメモリから追い出される時のキャッシュメモリとメインメモリ間のトラフィックを減少することができる。また、前記処理は途中放棄を許すため、他のアクセス手段とメインメモリ間のトラフィックを妨げることがない。従って、これらの効果によりシステム全体としてのトラフィックを軽減することができ、システムの性能向上を図ることができる。

【図面の簡単な説明】

【図1】本発明のキャッシュメモリ制御方法における第1の実施例を示すフローチャートである。

【図2】キャッシュメモリにおける状態説明図である。

【図3】本発明のキャッシュメモリ制御方法における第1の実施例を実現するためのシステム構成図である。

【図4】本発明のキャッシュメモリ制御方法における書き戻しバスコマンドのフォーマット構成図である。

【図5】本発明のキャッシュメモリ制御方法における第2の実施例を実現するためのシステム構成図である。

【図6】本発明のキャッシュメモリ制御方法における第2の実施例を示すフローチャートである。

【符号の説明】

1 プロセッサ

2、15 キャッシュメモリ

2a、15a キャッシュ管理テーブル

2b、15b 自律的書き戻し制御部

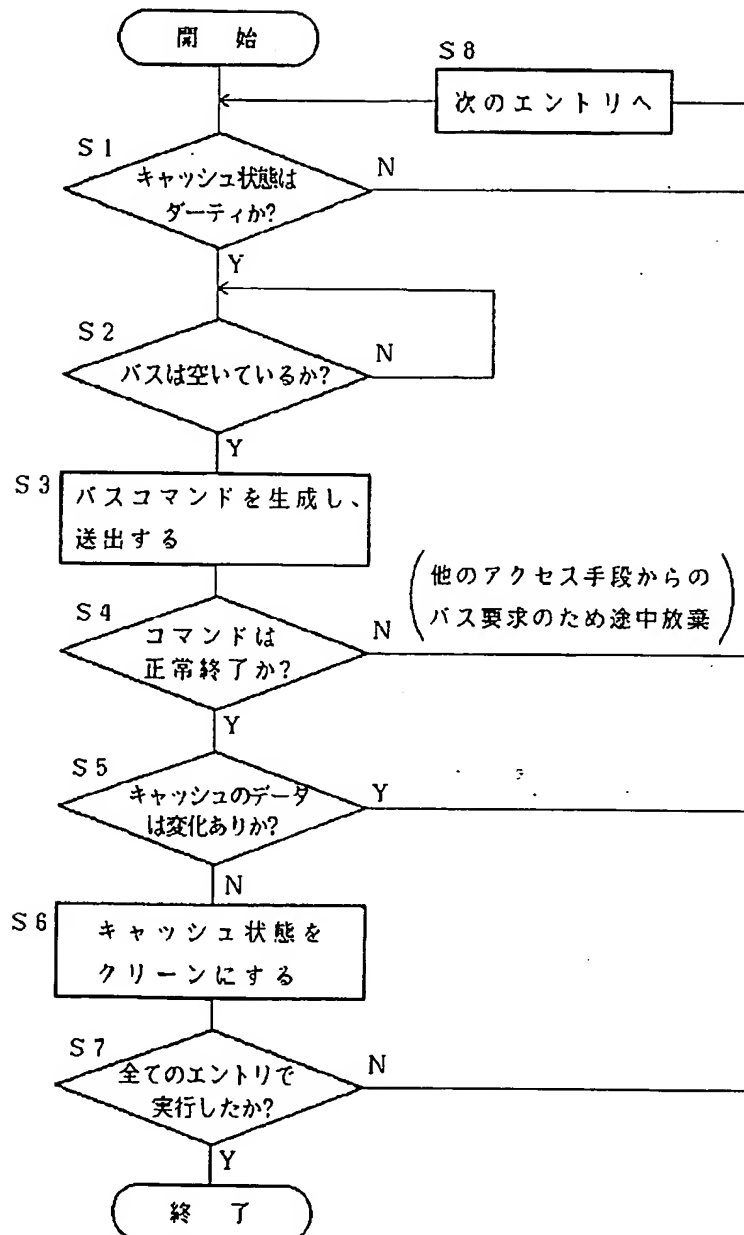
3、12 メインメモリ

4 I/Oプロセッサ

5、13 共有バス

11-1~11-n プロセッサエレメント

【図1】



本発明方法における第1の実施例のフローチャート

【図2】

I	Invalid	無効
CE	Private Clean (Clean Exclusive)	変更されていない
CS	Shared Clean	変更されていない
DE	Dirty (Dirty Exclusive)	変更されている

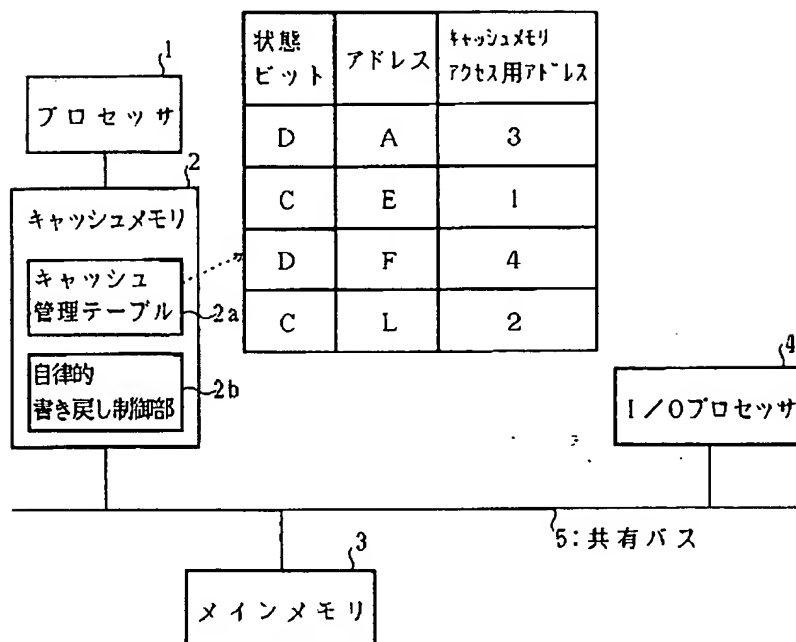
キャッシュ状態説明図

【図4】

制御 フラグ	アドレス	データ
-----------	------	-----	-------

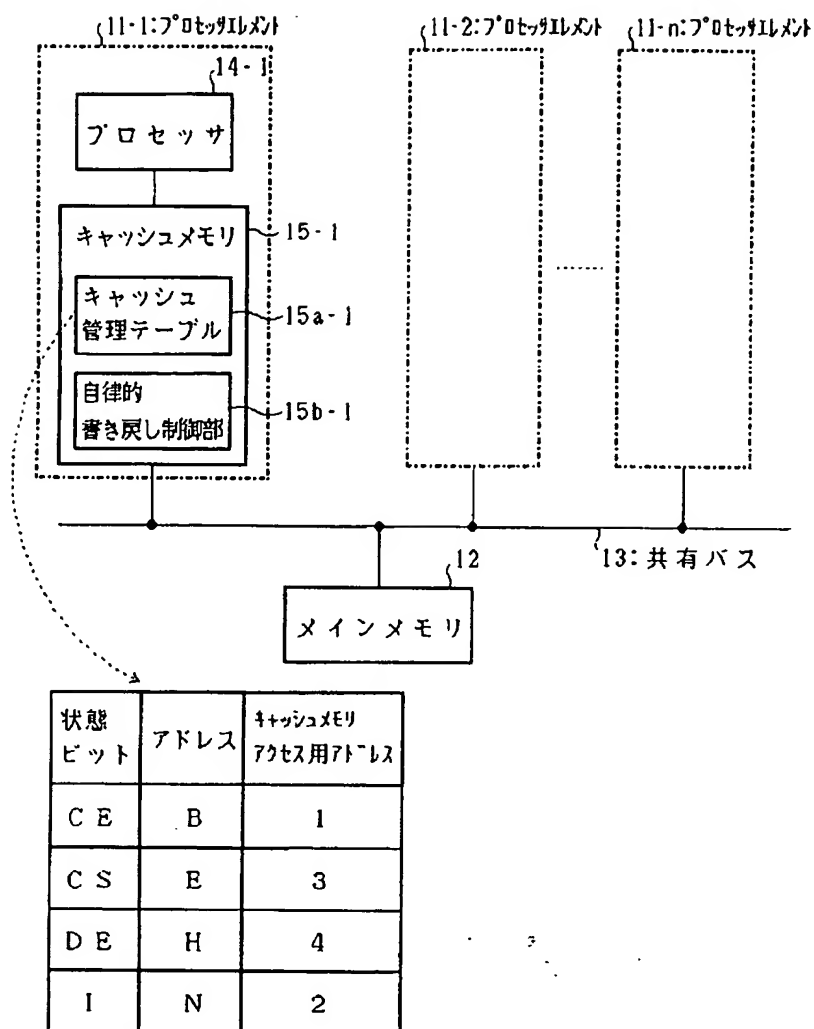
本発明方法における書き戻しバスコマンド

【図3】



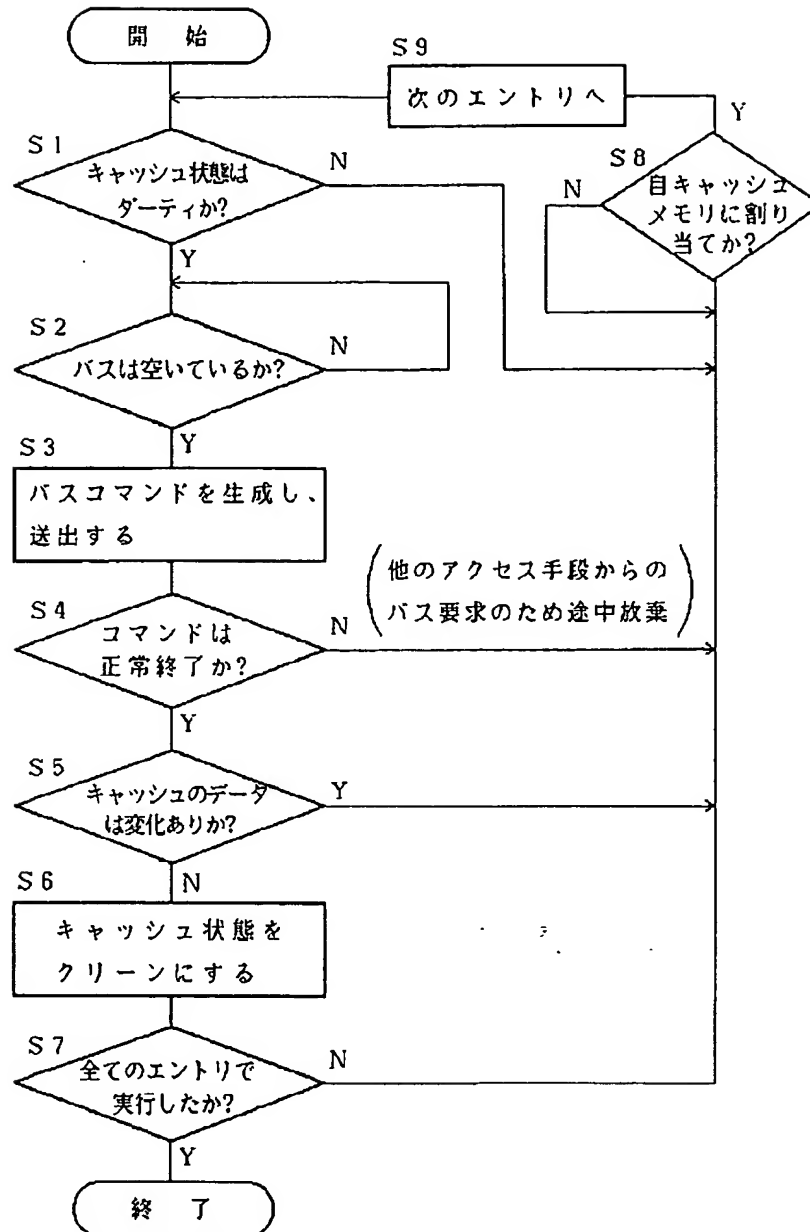
第1の実施例を実施するためのシステム構成

【図5】



第2の実施例を実現するためのシステム構成

【図6】



本発明方法における第2の実施例のフローチャート